One Identity Manager 8.1.3

Administration Guide for Connecting to Native Databases through Database Systems Integration Module

One Identity Manager Administration Guide for Connecting to Native Databases through Database Systems Integration Module
Updated - June 2020
Version - 8.1.3

Contents

1

Database Systems Integration Module

Database Systems Integration (DSI) module provides synchronization project templates, which enable you to to connect to multiple Database engines using the One Identity Manager native database connector. The DSI module project templates help in configuring synchronization components such as, object type mapping, workflows, and startup configurations.

The DSI module supports connection to the following database engines:

- Oracle Database
- Microsoft SQL database
- SAP HANA Database
- MySQL Database

  This document provides detailed information about the DSI module and its customizable templates, which provides mappings, workflows, and startup configurations to synchronize and provision database objects using native database connector.

Installing the Database Systems Integration module

This chapter provides information about the DSI module installation on supported operating systems. For more information on the supported operating systems, see *System Requirements* in the *One Identity Manager Release Notes for Database Systems Integration Module 8.1.3*.

Prerequisites to install the Database Systems Integration module

To install the Database Systems Integration (DSI) module on the system, ensure that the One Identity Manager 8.1.3 is already installed on the system. For more information on installing the latest version of the One Identity Manager, see the *Installing One Identity Manager Components* section in the *One Identity Manager Installation Guide*.

Installing the Database Systems Integration module

The Database Systems Integration (DSI) module installation is similar to the installation procedures of other One Identity Manager modules. To install the DSI module, One Identity Manager 8.1.3 is required.

For information on installing the DSI module, see the *Installing One Identity Manager Components* section in the *One Identity Manager Installation Guide*.

3

Working with the Oracle database template

This chapter provides information about the Oracle database template provided in the Database Systems Integration (DSI) module that is used to create mappings, workflows, startup configurations, and synchronize objects.

Features supported

The following features are supported in the Oracle database template:

- Read all user accounts and their attributes
- Read all user roles
- Read all user profiles
- Read all tablespaces
- Read user role membership
- Read role membership
- Create and delete operations for users
- Change password for users
- Lock or unlock users
- Change role membership
- Change user role membership

Prerequisites to configure the Oracle database

Ensure that the following prerequisites are met before configuring Oracle database:

- Ensure that the information about the following attributes are available.
  - Hostname
  - Service name
  - Port
  - Username
  - Password
- The native database configuration server function must be enabled on the Designer for the Job server.

For more information on assigning server function, refer to the *One Identity Manager Configuration Guide*.

Creating a synchronization project

The following procedure describes how to create and configure synchronization projects to work with the Oracle database.

To create a synchronization project

1. Open the Synchronization Editor.
2. On the **Start** page, select **Start a new Synchronization Project**.
3. On the **Choose target system page**, select the **Native Database Connector** check box and click **Next**.
4. Select **Create New System Connection**.
5. Select **Oracle Database**.
6. By default, the **Direct Access** check box is selected. If not, select the check box and proceed further.
7. Define the connection parameters as mentioned here:
   - **Server**: Provide the Hostname.
   - **Port**: Provide the port to the database. By default, the port text field is populated.
   - **Service Name**: Provide the service name of the Oracle database.

- **User**: SYSTEM.
- **Password**: Provide the password created during Oracle database installation.

8. Click **Extended** and set **Connect Mode** to **SYSDBA**.
9. Provide the display name and the identifier.
10. Select the `DSI_ORACLESQL_Configuration.xml` configuration file available at DSI/ConfigFiles in the OneIM installation folder, and click **Next**.
11. Click **Finish**.

    The schema loading window is displayed.
12. After the schema is loaded, the **Template Selection** page is displayed.
13. Select **Oracle Database Template** and click **Next**.

    Follow the on-screen instructions until the **Provisioning Capability** window is displayed.
14. Select the synchronization server and click **Finish**.

    The synchronization project is created successfully.

After creating the synchronization project, set the variable value in the **DefaultUserPassword** field.

Mapping sets and object types

The Oracle database template has the following mapping sets:

- **Tablespace**: Logical storage units that store data of all the databases.
- **Users**: Users of the database.
- **Roles**: Provides database security at a basic level.
- **Profile**: A collection of parameters that set limits on the database resources.
- **RoleInRole**: Role membership.
- **UserInRoles**: User role membership.
- **UserHasTablespace**: Permanent tablespace assigned to users.
- **UserHasTempTablespace**: Temporary tablespace assigned to users.

  NOTE: Before updating the default mapping sets, you should save the backup of the One Identity Manager database.

Mapping object types between Oracle database and One Identity Manager tables

The following table describes how to map object types in the Oracle database to the corresponding UNS tables in One Identity Manager.

Table 1: Mapping table for Oracle object types

Mapping set name	Oracle objects	One Identity Manager table
Tablespaces	sys.dba_tablespaces	UNSItemB
Users	sys.dba_users	UNSAccountB
Roles	sys.dba_roles	UNSGroupB
Profiles	sys.dba_profiles	UNSContainerB
RoleInRole	sys.dba_role_privs (GROUPINGROUP)	UNSGroupBInUNSGroupB
UserInRole	sys.dba_role_privs (USERINROLE)	UNSAccountBInUNSGroupB (effective assignments)
UserHasTablespace	sys.dba_users	UNSAccountBHasUNSItemB (Tablespace)
UserHasTempTablespace	sys.dba_users	UNSAccountBHasUNSItemB (Temporary Tablespace)

Initializing synchronization

Initial synchronization initializes the One Identity Manager tables with the Oracle data. Oracle users, roles, tablespaces, user role membership, and role membership are synchronized into the One Identity Manager tables.

To run the initial synchronization

1. In the **Synchronization Editor**, open the synchronization project.
2. In the Navigation pane, select **Start up Configurations**.
3. Click **Execute**.
4. Confirm the security prompt with **Yes**.

   The synchronization workflow is completed successfully. The synchronized objects can be viewed in the Manager.

Provisioning workflow

The **Provisioning** workflow is used to create, update, update role membership, and delete Oracle users.

Creating an Oracle user

The following procedure describes how to create Oracle users on the Oracle database.

To create an Oracle user

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**. To view the target system name, expand the required target system root.
3. Click **Add**.
4. Provide the relevant information for the new user.

   NOTE: The **Login name** for the user must be unique.
5. Provide a password in the **Password** text field. If not provided, the value from the **DefaultUserPassword** variable is used.
6. Click **Save**.

   The user is provisioned in the Oracle Database.

Updating attributes of an Oracle user

The following procedures describes how to create the Oracle users on Oracle database.

To change the password:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**. To view the target system name, expand the required target system root.
3. Provide the new password in the **Password** text field.
4. Click **Save**.

   The password is updated.

To lock or unlock users:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the user to be updated.
4. Select the **User account is disabled** check box to lock or unlock the user.

   The user is locked or unlocked based on the action.

To change role membership:

1. Open the Manager.
2. Navigate to **Custom target systems | <*required target system root*> | User accounts**.
3. In **Custom target system**, select the user to be updated.
4. From the **Assign Groups** section, add or remove groups to update the role membership.

   The role membership is updated.

Deleting an Oracle user

The following procedure describes how to delete an Oracle user.

To delete an Oracle user

1. Open the Manager.
2. Navigate to **Custom target systems | <*required target system root*> | User accounts**.
3. In **Custom target systems**, select the user and click **Delete**.

The user account is disabled immediately on the Oracle database based on deferred deletion value of the **UNSAccountB** table in One Identity Manager.

Updating role membership

The following describes how to create Oracle users on the Oracle database.

To update role membership

1. Open the Manager.
2. Navigate to **Custom target systems | <*required target system root*> | User accounts**.
3. In **Custom target systems**, select the **Group** to be updated.
4. From the **Assign Groups** section, add or remove groups to update the role membership.

   The role membership is updated.

4

Working with Microsoft SQL Server templates

This chapter provides information about the Microsoft SQL (MS SQL) database templates for server level and database level provided in the Database Systems Integration (DSI) module that is used to create mappings, workflows, startup configurations, and synchronize objects.

The server-level template is used to connect to the master database to manage server objects. The database-level template is used to connect to specific databases to manage database objects.

Features supported

The following features are supported in the MS SQL templates.

Features supported for MS SQL Server-level template

- Read all login accounts and their attributes.
- Read all server roles.
- Read all login role membership.
- Read all role membership.
- Create and delete operations for login.

- Change password for logins.
- Lock or unlock logins.
- Change role membership.
- Change login role membership.

Features supported for MS SQL Database-level template

- Read all user accounts and their attributes.
- Read all database roles or application roles.
- Read all user role membership.
- Read all role membership.
- Create and delete operations for users.
- Change password for users.
- Lock or unlock users.
- Change role membership.
- Change user role membership.

Prerequisites to configure the MS SQL database

Ensure that the following prerequisites are met before configuring the MS SQL database:

- Before creating a synchronization project with the native database connector, ensure that the information about the following attributes is available:
  - MS SQL Server Name
  - Username
  - Password
  - Database that is to be connected
- The native database configuration server function must be enabled on the Designer for the Job server.

For more information on assigning server function, refer to the *One Identity Manager Configuration Guide*.

Creating a synchronization project

The synchronization project for MS SQL is classified into server level and database level.

Synchronization project for server level

The following procedure describes how you create and configure synchronization projects for server-level projects in Microsoft SQL.

To create synchronization project

1. Open the Synchronization Editor.
2. On the **Start** page, select **Start a new Synchronization Project**.
3. On the **Choose target system page**, select the **Native Database Connector** check box and click **Next**.
4. Select **Create New System Connection**.
5. Select **SQL Server**.
6. Provide the name of the SQL server required to establish a connection in the **SQL Server Name** field.
7. Provide the **Username** and **Password** for the privileged service account or an account that has the **sysadmin** server role.
8. Select **master** database to create the project for managing server-level objects.
9. Provide the display name and the identifier.
10. Select the `DSI_MSSQL_ServerLevel_Configuration.xml` configuration file available at `DSI/ConfigFiles` in the DSI installation folder and click **Next**.
11. Review the preferences and click **Next** until the **Save Configuration** window is displayed. Ensure that the configuration field is empty.
12. Click **Next**.
13. Review the preferences and click **Finish**.

    The schema loading window is displayed.
14. After the schema is loaded, the **Template Selection** page is displayed.
15. Select **MSSql Server Level Template** and click **Next**.

    Follow on-screen instructions until the **Provisioning Capability** window is displayed.
16. Select the synchronization server and click **Finish**.

    The synchronization project is created successfully.

After the creation of the synchronization project, set the variables value in the **DefaultUserPassword** field.

Synchronization project for database level

The following procedure describes how to create and configure synchronization projects for database-level projects in Microsoft SQL.

To create a synchronization project

1. Open the Synchronization Editor.
2. On the **Start** page, select **Start a new Synchronization Project**.
3. On the **Choose target system page**, select the **Native Database Connector** check box and click **Next**.
4. Select **Create New System Connection**.
5. Select **SQL Server**.
6. Provide the name of the SQL server required to establish a connection in the **SQL Server Name** field.
7. Provide the **Username** and **Password** for the privileged service account or an account that has the **sysadmin** server role.
8. Select the required database to manage database-level principals.
9. Provide the display name and the identifier.
10. Select the `DSI_MSSQL_DBLevel_Configuration.xml` configuration file available at `DSI/ConfigFiles` in the DSI module and click **Next**.
11. Review the preferences and click **Next** until the **Save Configuration** window is displayed. Ensure that the configuration field is empty.
12. Click **Next**.
13. Review the preferences and click **Finish**.

    The schema loading window is displayed.
14. After the schema is loaded, the **Template Selection** page is displayed.
15. Select **MSSql DB Level Template** and click **Next**.

    Follow on-screen instructions until the **Provisioning Capability** window is displayed.
16. Select the synchronization server and click **Finish**.

    The synchronization project is created successfully.

After the creation of the synchronization project, set the variables value in the **DefaultUserPassword** field.

Connecting to multiple databases through a single synchronization project

Database Systems Integration (DSI) provides an additional template for the Microsoft SQL database that allows it to connect to multiple databases through a single synchronization project in Microsoft SQL at database-level.

The ability to connect to multiple databases through a single synchronization project simplifies the process of maintaining multiple database connections, as the databases share a single set of configurations including mappings, workflows, and data operations.

To create connections to multiple databases through a single synchronization project

1. In the Synchronization Editor, open the synchronization project for MS SQL database-level connection.
2. In the **configuration** pane, navigate to the base object and click the wizard icon to set up a connection to the database.
3. Follow the steps specified in the Synchronization project for database level section to add base objects.
4. Commit the changes to the database.

   A new startup configuration gets created for the base object connection.

Mapping sets and object types

The Microsoft SQL template is classified into a server-level template and a database-level template. The mapping sets and the corresponding object types are different for server-level and database-level templates.

Mapping sets and object types for the server-level template

The Microsoft SQL server-level template has the following mapping sets:

- **Logins**: Maps all logins thataccess the SQL server, such as, **SQL_LOGIN**, **WINDOWS_LOGIN**, and **WINDOWS_GROUP**, into One Identity Manager.
- **ServerRoles:** Maps all server-level roles and user-defined server roles into One Identity Manager.
- **LoginHasRole**: Maps all login and their role membership into One Identity Manager.
- **ServerRoleinServerRole**: Maps all roles and their role membership into One Identity Manager.

NOTE: Before updating the default mapping sets, you should save the backup of the One Identity Manager database.

Mapping object types between SQL tables and One Identity Manager tables

The following table describes here briefs about the mapping of server object types in MS SQL Server to the corresponding UNS tables of One Identity Manager.

Table 2: Mapping table for MS SQL server-level object types

MS SQL server-level object types		
Mapping set name	**MS SQL objects**	**One Identity Manager table**
Login	sys.server_principals	UNSAccountB
ServerRoles	sys.server_principals	UNSGroupB
LoginHasRoles	sys.server_role_ members	UNSAccountBInUNSGroupB
ServerRoleInServerRole	sys.server_role_ members	UNSGroupBInUNSGroupB

Mapping sets and object types for the database-level template

The Microsoft SQL database-level template has the following mapping sets:

- **Users**: Maps all users to access the databases such as, **SQL_USER_WITHOUT_ LOGIN**, **SQL_USER_WITH_LOGIN**, **SQL_USER_WITH_PASSWORD** (applicable only for contained databases), **WINDOWS_USER** and **WINDOWS_GROUP** into One Identity Manager.
- **Roles**: Maps all database and application roles for the database into One Identity Manager.
- **UserHasRoles**: Maps all users and their role membership into One Identity Manager.
- **RoleInRole**: Maps all roles and their role membership into One Identity Manager.
- **UsersExtended**: Maps all user properties for provisioning from One Identity Manager to MS SQL.

  NOTE: Before updating the default mapping sets, you should save the backup of the One Identity Manager database.

Mapping object types between SQL tables and One Identity Manager tables

The following table here describes mapping database object types in Microsoft SQL Server to the corresponding UNS tables of One Identity Manager.

Table 3: Mapping table for MS SQL database-level object types

MS SQL database-level object types		
Mapping set name	**MS SQL objects**	**One Identity Manager table**
Users	sys.database_principals	UNSAccountB
Roles	sys.database_principals	UNSGroupB
UserHasRoles	sys.database_role_ members	UNSAccountBInUNSGroupB
RoleInRole	sys.database_role_ members	UNSGroupBInUNSGroupB
UsersExtended	sys.sysusers	UNSAccountB

Initial synchronization

Initial synchronization is used to initialize the One Identity Manager tables with the MS SQL data. Users, logins, roles, user role membership, login role membership, and role membership are synchronized into the One Identity Manager tables.

To run the initial synchronization

1. In the **Synchronization Editor**, open the synchronization project.
2. In the Navigation section, select **Start up Configurations**.
3. Click **Execute**.
4. Confirm the security prompt with **Yes**.

   The synchronization workflow is completed successfully. The synchronized objects can be viewed in the Manager.

Provisioning workflow

The **Provisioning** workflow is used to create, update, and delete Microsoft SQL login and users.

Provisioning workflow for server-level objects

The **Provisioning** workflow for server-level objects is used to create, update, delete Microsoft SQL login, and update server-role membership.

Creating a Microsoft SQL login

You can create **SQL LOGIN**, **WINDOWS LOGIN**, and **WINDOWS GROUP** types of logins in the MS SQL Server.

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**. To view the target system name, expand the required target system root.
3. Click **Add**.
4. Provide the relevant information for the new login.

   NOTE:

   - For **WINDOWS LOGIN** and **WINDOWS GROUP**, the user name must be in ***<domain\username>*** format.
   - The **Login name** for the user must be unique.
5. Provide a password in the **Password** text field. If not provided, the value from the **DefaultUserPassword** variable is used.
6. Select one of the following options, **SQL LOGIN**, **WINDOWS LOGIN**, or **WINDOWS GROUP** from the **AccountType** drop-down list to proceed further with the process.
7. Click **Save**.

   The user is provisioned in MS SQL successfully.

Updating attributes of a Microsoft SQL login

The following procedures provides information about the process to change password, lock users, unlock users, and change membership.

To change the password (applicable only to SQL LOGIN):

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the login to be updated.
4. Provide the new password in the **Password** text field.

5. Click **Save**.

   The password is updated.

To lock or unlock users:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the login to be updated.
4. Select the **User account is disabled** check box to lock or unlock the user.

   The user is locked or unlocked based on the action.

To change role membership:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the login to be updated.
4. From the **Assign Groups** section, add or remove groups to update the role membership.

   The role membership is updated.

Deleting a Microsoft SQL login

The following procedure describes how to delete an SQL login.

To delete a MS SQL login

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the login and click **Delete**.

The user account is disabled immediately on the MS SQL database based on deferred deletion value of the **UNSAccountB** table in One Identity Manager.

Updating server-role membership

The following topic briefs about the procedure to update the server-role membership.

To update server-role membership

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the **Group** to be updated.
4. From the **Assign Groups** section, add or remove groups to update the server-role membership.

   The server-role membership is updated.

Provisioning workflow for database-level objects

The **Provisioning** workflow for database-level objects is used to create, update, delete MS SQL users, and update database-role membership.

Creating a Microsoft SQL user

You can create **SQL USER WITHOUT LOGIN**, **SQL USER WITH LOGIN**, **WINDOWS USER**, **WINDOWS GROUP**, and **SQL USER WITH PASSWORD** types of users in MS SQL Server.

IMPORTANT:

- The SQL user is created and automatically mapped to the login with the same username if it exists, or else a new login is created.
- A login is not required to access the SQL server for **SQL USER WITH PASSWORD**. These users are only applicable to a contained database.

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**. To view the target system name, expand the required target system root.
3. Click **Add**.
4. Provide the relevant information for the new user.

   IMPORTANT:

   - For **WINDOWS LOGIN** and **WINDOWS GROUP**, the user name must be in the ***<domain\username>*** format.
   - The **Login name** for the user must be unique.
5. Provide a password in the **Password** text field. Entering a password is applicable only for **SQL USER WITH LOGIN** and **SQL USER WITH PASSWORD** users. If not provided, the value from the **DefaultUserPassword** variable is used.NOTE: For

SQL USER WITH LOGIN, the password text field is used only for login creation.

NOTE: For **SQL USER WITH LOGIN**, the password text field is used only for login creation.

6. Select one of the following options, **SQL USER WITHOUT LOGIN**, **SQL USER WITH LOGIN**, **WINDOWS USER**, **WINDOWS GROUP**, or **SQL USER WITH PASSWORD** from the **AccountType** drop-down list to proceed with the process.
7. Click **Save**.

   A MS SQL user is created successfully.

Updating attributes of a Microsoft SQL user

The following topic briefs about the procedure to change password, lock user, unlock users, and change role membership.

To change the password:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the user to be updated.
4. Provide the new password in the **Password** text field.
5. Click **Save**.

   The password is updated.

To lock or unlock users:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the user to be updated.
4. Select the **User account is disabled** check box to lock or unlock the user.

   The user is locked or unlocked based on the action.

To change role membership:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the user to be updated.
4. From the **Assign Groups** section, add or remove groups to update the role

membership.

The role membership is updated.

Deleting a Microsoft SQL user

The following procedure describes how to delete an SQL user.

To delete a MS SQL user

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the user and click **Delete**.

The user account is disabled immediately on the MS SQL database based on deferred deletion value of the **UNSAccountB** table in One Identity Manager.

Updating database role membership

The following describes the procedure to update database role membership.

To update database role membership

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the **Group** to be updated.
4. From the **Assign Groups** section, add or remove groups to update the database-role membership.

   The database-role membership is updated.

5

Working with the SAP HANA database template

This chapter provides information about the SAP HANA database template provided in the Database Systems Integration module that is used to create mappings, workflows, startup configurations, and synchronize objects.

Features supported

The following features are supported in the SAP HANA database.

- Read all user accounts and their attributes.
- Read all user roles.
- Read all user role membership.
- Read role membership.
- Create and delete operations for users.
- Change password.
- Lock or unlock user.
- Change role membership.
- Change user role membership.

Prerequisites to configure the SAP HANA database

Ensure that the following prerequisites are met before configuring the SAP HANA database.

- Before creating a new synchronization project on the SAP HANA database, install the SAP HANA client on a system that is set as the Job server.
- Ensure that the information about the following attributes of the SAP HANA target system are available.
    - Hostname
    - Port number to the database
    - Username
    - Password
- The native database configuration server function must be enabled on the Designer for the Job server.

For more information on assigning server function, refer the *One Identity Manager Configuration Guide*.

Creating a synchronization project

The following procedure explains how to create and configure synchronization projects in the SAP HANA database.

Creating a synchronization project

1. Open the Synchronization Editor.
2. On the **Start** page, select **Start a new Synchronization Project**.
3. On the **Choose target system page**, select the **Native Database Connector** check box and click **Next**.
4. Select **Create New System Connection**.
5. Select **SAP HANA** and choose **SAP HANA Data Provider** from the drop-down menu.
6. In the **Server** field, enter the hostname and the portname to the database created on SAP HANA in the following format: **Hostname:portnumber**. For example, **<*hostname*>:39013** to connect to SYSTEMDB and **<*hostname*>:39015** to connect to HXE default database.
7. Provide the **Username** and **Password** to the database.
8. Provide the display name and the identifier.

9. Select the `DSI_SAPSQL_Configuration.xml` configuration file available at `DSI/ConfigFiles` in the One Identity Manager installation folder, and click **Next**.
10. Review the preferences and click **Next** until the **Save Configuration** window is displayed. Ensure that the configuration field is empty.
11. Click **Next**.
12. Review the preferences and click **Finish**.

    The schema loading window is displayed.
13. After the schema is loaded, the **Template Selection** page is displayed.
14. Select the **SAP HANA** template and click **Next**.

    Follow on-screen instructions until the **Provisioning Capability** window is displayed.
15. Select the synchronization server and click **Finish**.

    The synchronization project is created successfully.

After the creation of the synchronization project, set the variables value in the **DefaultUserPassword** field.

Connecting to multiple databases through a single synchronization project

The Database Systems Integration (DSI) module provides an additional template for the SAP HANA database that allows you to connect to multiple databases through a single synchronization project in SAP HANA.

The ability to connect to multiple databases through a single synchronization project simplifies the process of maintaining multiple databases connections, as the databases share a single set of configurations including mappings, workflows, and data operations.

To create connections to multiple databases through a single synchronization project

1. In the Synchronization Editor, open the synchronization project for SAP HANA connection.
2. In the **configuration** pane, navigate to the base object and click the wizard icon to setup a connection to the database.
3. Follow the steps specified in the Creating a synchronization project for SAP HANA database to add base objects.
4. Commit the changes to the database.

   A new startup configuration is created for the base object connection.

Mapping sets and object types

The SAP HANA database template has mapping sets for the following SAP HANA object types:

- User
- Roles
- Granted Roles

NOTE: Before updating the default mapping sets, you should save the backup of the One Identity Manager database.

Mapping object types between the SAP HANA database and One Identity Manager tables

The following table describes the mapping object types in the SAP HANA database to the parameters in the UNS tables of One Identity Manager.

Table 4: Mapping table for SAP HANA object types

Mapping set name	Object type	One Identity Manager table
UsersMapping	SYS.USERS	UNSAccountB
RolesMapping	SYS.ROLES	UNSGroupB
UserHasRoleMapping	SYS.GRANTED_ROLES	UNSAccountBInUNSGroupB
RoleHasRoleMapping	SYS.GRANTED_ROLES	UNSGroupBInUNSGroupB

Initial synchronization

Initial synchronization is used to initialize the One Identity Manager tables with SAP HANA data. SAP HANA users, roles, user role membership, and role membership are synchronized into the One Identity Manager tables.

To run the initial synchronization

1. In the **Synchronization Editor**, open the synchronization project.
2. In the Navigation section, select **Start up Configurations**.
3. Click **Execute**.
4. Confirm the security prompt with **Yes**.

The synchronization workflow is completed successfully. The synchronized objects can be viewed in the Manager.

Provisioning workflow

The **Provisioning** workflow is used to create, update, delete SAP HANA users, and update role membership.

Creating a SAP HANA user

The following procedure describes how to create a SAP HANA user.

To create a SAP HANA user

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**. To view the target system name, expand the required target system root.
3. Click **Add**.
4. Provide the relevant information for the new user.

   NOTE: The **Login name** for the user must be unique.
5. Provide a password in the **Password** text field. If not provided, the value from the **DefaultUserPassword** variable is used.

   NOTE:Select **RESTRICTED** from the **AccountType** drop-down list to create restricted users in SAP HANA.
6. Click **Save**.

   The SAP HANA user is provisioned successfully.

Updating attributes of an SAP HANA user

The following topic briefs about the process of updating attributes of a SAP HANA user.

To change the password:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In the **Custom target system**, select the user to be updated.

4. Provide the new password in the **Password** text field.
5. Click **Save**.

   The password is updated.

To lock or unlock users:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the user to be updated.
4. Select the **User account is disabled** check box to lock or unlock the user.

   The user is locked or unlocked based on the action.

To change role membership:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the user to be updated.
4. From the **Assign Groups** section, add or remove groups to update the role membership.

   The role membership is updated.

Deleting a SAP HANA user

The following procedure describes how to delete a SAP HANA user.

To delete a SAP HANA user

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the user and click **Delete**.

The user account is disabled immediately on the SAP HANA database based on deferred deletion value of the **UNSAccountB** table in One Identity Manager.

Updating role membership

The following describes the process of updating the role membership of a SAP HANA user.

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the **Group** to be updated.
4. From the **Assign Groups** section, add or remove groups to update the role membership.

   The role membership is updated.

Working with the MySQL database template

This chapter provides information about the MySQL database template provided in the Database Systems Integration module that is used to create mappings, workflows, startup configurations, and synchronize objects.

Features supported

The following features are supported in the MySQL database.

- Read all user accounts and their attributes
- Read all user privileges
- Read user assigned privileges

The following features are supported in the MySQL database user

- Create, read, update, and delete users
- Change password
- Lock or unlock user
- Change privilege membership

Prerequisites to configure the MySQL database

Ensure that the following prerequisites are met before configuring the MySQL database.

- Before creating a new synchronization project on the MySQL database, install MySQL client on a system that is set as the Job server and the client system.

- Download and install the MySQL .NET connector version 6.9.12. Ensure that the MySQL.data.dll is available on the Job server and in the location where the One Identity Manager is installed.
- Ensure that the information about the following attributes of the MySQL target system are available.
    - Hostname
    - Username
    - Password
    - Database- by default, mysql is the fixed entry.
- The native database configuration server function must be enabled on the Designer for the Job server.
- The Information on assigned privileges for MySQL cannot be retrieved directly using the existing schema. You must create custom views in the MySQL server using the sql script file MYSQL_SERVER_VIEWS.sql available in `DSI/ConfigFiles` location. To create custom views run the following queries in the mysql server before creating synchronization project for MySQL:
    - `CREATE VIEW mysql.all_privileges AS SELECT DISTINCT privilege_type from information_schema.USER_PRIVILEGES` query.
    - `CREATE VIEW mysql.user_privileges AS SELECT * from information_schema.USER_PRIVILEGES` query.

For more information on assigning server function, refer the *One Identity Manager Configuration Guide*.

Creating a synchronization project

The following procedure explains how to create and configure synchronization projects in the MySQL database.

Creating a synchronization project

1. Open the Synchronization Editor.
2. On the **Start** page, select **Start a new Synchronization Project**.
3. On the **Choose target system page**, select the **Native Database Connector** check box and click **Next**.
4. Select **Create New System Connection**.
5. Select **MySQL** from the drop-down list.
6. In the **Server** field, enter the hostname in the following format: **Hostname:portnumber**.
7. Provide the **Username** and **Password** to the database.
8. Select mysql in the drop-down list for the **Database** field.

9. Provide the display name and the identifier.
10. Select the `DSI_MYSQL_Configuration.xml` configuration file available at `DSI/ConfigFiles` in the One Identity Manager installation folder, and click **Next**.
11. Review the preferences and click **Next** until the **Save Configuration** window is displayed. Ensure that the configuration field is empty.
12. Click **Next**.
13. Review the preferences and click **Finish**.

    The schema loading window is displayed.
14. After the schema is loaded, the **Template Selection** page is displayed.
15. Select the **MySQL** template and click **Next**.

    Follow on-screen instructions until the **Provisioning Capability** window is displayed.
16. Select the synchronization server and click **Finish**.

    The synchronization project is created successfully.

After the creation of the synchronization project, set the variables value in the **DefaultUserPassword** field.

Mapping sets and object types

The MySQL database template has mapping sets for the following MySQL object types:

- User
- All_Privileges
- User_Privileges

  NOTE: Before updating the default mapping sets, you should save the backup of the One Identity Manager database.

Mapping object types between the MySQL database and One Identity Manager tables

The following table describes the mapping object types in the MySQL database to the parameters in the UNS tables of One Identity Manager.

Table 5: Mapping table for MySQL object types

Mapping set name	Object type	One Identity Manager table
UsersMapping	Mysql.user	UNSAccountB
PrivilegesMapping	Mysql.all_	UNSGroupB

	privileges	
UserHasPrivilegeMapping	Mysql.user_ privileges	UNSAccountBInUNSGroupB

Initial synchronization

Initial synchronization is used to initialize the One Identity Manager tables with MySQL data. MySQL users, roles, user membership, and role membership are synchronized into the One Identity Manager tables.

To run the initial synchronization

1. In the **Synchronization Editor**, open the synchronization project.
2. In the Navigation section, select **Start up Configurations**.
3. Click **Execute**.
4. Confirm the security prompt with **Yes**.

   The synchronization workflow is completed successfully. The synchronized objects can be viewed in the Manager.

Provisioning workflow

The **Provisioning** workflow is used to create, update, delete MySQL users, and update role membership.

Creating a MySQL user

The following procedure describes how to create a MySQL user.

To create a MySQL user

1. Open the Manager.
2. Navigate to **Custom target systems | <*required target system root*> | User accounts**. To view the target system name, expand the required target system root.
3. Click **Add**.
4. Provide the relevant information for the new user.

   NOTE: The **Login name** for the user must be unique.

5. Provide a password in the **Password** text field. If not provided, the value from the **DefaultUserPassword** variable is used.
6. Optionally provide the hostname in the **Hostname** field.
7. Click **Save**.

   The MySQL user is provisioned successfully.

Updating attributes of MySQL user

The following topic briefs about the process of updating attributes of a MySQL user.

To change the password:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In the **Custom target system**, select the user to be updated.
4. Provide the new password in the **Password** text field.
5. Click **Save**.

   The password is updated.

To lock or unlock users:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the user to be updated.
4. Select the **User account is disabled** check box to lock or unlock the user.

   The user is locked or unlocked based on the action.

To change privilege membership:

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target system**, select the user to be updated.
4. From the **Assign Groups** section, add or remove groups to update the role membership.

   The role membership is updated.

Deleting a MySQL user

The following procedure describes how to delete a MySQL user.

To delete a MySQL user

1. Open the Manager.
2. Navigate to **Custom target systems | *<required target system root>* | User accounts**.
3. In **Custom target systems**, select the user and click **Delete**.

The user account is disabled immediately on the MySQL database based on deferred deletion value of the **UNSAccountB** table in One Identity Manager.

6

User scenario

This chapter provides information about user scenarios in the Database Systems Integration (DSI) module.

Provisioning logins in MS SQL

Provisioning of **Login Accounts** in MS SQL Server can be automated through account definitions in One Identity Manager. To automate the provisioning of **Login Accounts**, create an account definition for **UNSAccountB** and specify the MS SQL target system. Ensure that the account definition has the associated mappings with **UNSAccountB** for the successful processing of Login creation process.

Automating Login Account creation

The user accounts from the Active Directory (AD) are synchronized using the Active Directory connector for One Identity Manager. After synchronizing, the corresponding entries are displayed in the **Employee** section of the **Manager** tool. Assigning an account definition for the employee and saving operation creates a Windows Login for the Active Directory account in MS SQL Server.

Automating Login Account creation

1. Create an account definition for the MS SQL Server target system using the **Manager** tool.
2. Define an IT operating data mapping for **AccountType** column of the **UNSAccountB** table.

   IMPORTANT:

   - The template in the **Value Calculation** field must have the phrase `TSB_ITDataFromOrg` for the columns to be available for mapping. A comment line can also be added with `TSB_ITDataFromOrg`.

- To provision Windows Login, as a default value **WINDOWS_LOGIN** must be specified.
- **AccountName** in the **UNSAccountB** table is used by the MS SQL script to provision the Login

3. After the account definition is configured, assign it to the **Employee**.
4. Click **Save**.

   An entry is created in the **UNSAccountB** table and the process for provisioning of Login is triggered.

   NOTE:

   - If there is an issue in automating the Login Account creation, check the job queue for a detailed information about the issue.
   - Ensure that the correct values have been set in the **AccountType** and **LoginName** fields of the **UNSAccountB** table in the **Manager** tool.

7

Troubleshooting

Issues related to the use of this module range from server-related issues to Job server latencies. Some of the issues that can affect the use of this module are mentioned here:

- Network issues connecting the database instance with the Job server handling database synchronization and provisioning tasks.
- For details about the errors related to Microsoft SQL Server and Oracle databases, see the Job server logs.
- Detailed information on the errors for the SAP HANA **Synchronization** or **Provisioning** is not available in the Job Queue, as it uses SAP HANA client assemblies.
- Connectivity issues related to the instance unavailability or incorrect credentials being supplied while connecting to the database.
- Provisioning issues can be related to specific known issues of the target databases. For example, in Oracle database, **GLOBAL_AQ_USER_ROLE** cannot be assigned to a user object type.

For more information on known issues specific to the target databases, refer the *One Identity Manager Release Notes for Database Systems Integration Module*.

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit https://www.oneidentity.com/company/contact-us.aspx.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product